ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 1, Issue 10, December 2013

An User-Friendly and Improved Semantic-Based Web Service Discovery Approach Using Natural Language Processing Techniques

M. Deepa Lakshmi[1], Dr. Julia Punitha Malar Dhas[2]

Research Scholar, N.I. University, Kumaracoil, India[1]

Professor, Dept. of CSE, N.I. University, Kumaracoil, India[2]

ABSTRACT: Semantic Web services integrate the meaningful content of the Semantic Web with the business logic of Web services and thus enable industries and individuals to access these services. But as the number of available web services increase, there is a growing demand for a mechanism for effective retrieval of required services. We propose an improved Semantic Web Service Discovery method by combining functional similarity matching (using bipartite graph) and textual similarity matching. The framework searches through a set of annotated Web services for matching user query which can be represented in natural language, so that knowledge about semantic languages is not required by the user. We analyze the complexity of existing algorithms and present performance results, which show that our algorithm gives more flexibility and also performs well with respect to the existing algorithms.

Keywords: bipartite graph, OWL-S, matchmaking, functional similarity, textual similarity

I. Introduction

Today there exist a vast amount of tools that support and facilitate the development, deployment and invocation of Web Services, and there are virtually no limits as to what Web Services can do within their realm. With the rapid development of Web Services, the retrieval of relevant services has become a challenge. The keyword-based discovery mechanism using Universal Description, Discovery and Integration [1] (UDDI) and Web Services Description Language [2] (WSDL) is inefficient due to the retrieval of a large amount of irrelevant information. This is due to the lack of semantic information of the service. The Semantic Web initiative addresses this problem by creating a set of XML based documents and ontology. Semantic Web services integrate the meaningful content of the Semantic Web with the business logic of Web services and thus enable industries and individuals to access these services. But as the number of available web services increase, there is a growing demand for a mechanism for effective retrieval of required services.

An important component of the discovery process is the matchmaking algorithm. In order to overcome the limitations of existing syntax-based search, matchmaking algorithms based on semantic techniques have been proposed. Most of them are based on a logic-based semantic matchmaking approach proposed by Paolucci et al in [3]. This approach has certain limitations. At this juncture, improving the existing semantic-based web service discovery mechanism constitutes a vital step. Therefore, we propose an improved Semantic Web Service Discovery method for finding OWL-S (Web Ontology Language for Services) services using bipartite graph-based matching instead of Paolucci's algorithm (for functional similarity). Our proposed approach also performs a text-based similarity match.

One of the crucial steps in an efficient Web service search is to understand what users mean in their request, since the user's request is usually in the form of natural language. Most of the current popular service discovery algorithms expect that the user request (Query) be given only as an OWL-S query. But, our work searches through the set of annotated Web services for matches with a user query, which consists of just keywords, so that knowledge about semantic languages is not required. It permits user query to be specified in natural language and performs matching based on text similarity. The proposed method uses natural language processing (NLP) techniques to enhance user request and the description of the set of available services.

The proposed method consists of three steps:

- Enhancement of user request using NLP techniques

- Next step involves retrieving the relevant services based on their functional similarity (bipartite graph-based matching)
- Finally pre-processing of OWL-S services using NLP techniques and retrieving services based on their textual descriptions

Using OWL-S TC2, (OWL-S service retrieval test collection) we tested our improved algorithm and compared it with the existing approaches. The experimental results have shown that the proposed algorithm provides flexibility and could pick the most suitable advertised service corresponding to user's request from very similar ones and provide better recall rate and acceptable precision compared to the other approaches.

The rest of the paper is organized as follows: Section 2 explains some of the background details. Section 3 presents some of the related works. Section 4 gives an overview of proposed approach. Our evaluations are presented in Section 5. Finally, conclusion and future work are presented in Section 6.

II. Background

A. *Ontology*

Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts [4]. Ontologies are used to reason about the properties of a domain and may be used to define the domain. Upper ontologies are domain-independent ontologies, intended to be reused and extended for particular domain to form domain ontology

B. *OWL-S*

OWL-S is an ontology built on top of Web Ontology Language (OWL) developed by the DARPA DAML program. It replaces the former DAML-S ontology. OWL-S is an ontology, within the OWL-based framework of the Semantic Web, for describing Semantic Web Services. It will enable users and software agents to automatically discover, invoke, compose, and monitor Web resources offering services [5].

C. *WordNet*

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept [6]. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

D. *Jena*

Jena is a Java framework for building Semantic Web applications [7]. It provides extensive Java libraries for helping developers develop code that handles RDF, OWL and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL ontologies.

III. Related work

Majority of the current semantic web service discovery algorithms perform logic-based service profile matching, and are restricted to OWL-S services. The most influencing logic-based semantic matchmaking we are aware of is the Paolucci et al. algorithm [3], which has been cited extensively in subsequent proposals. In [3], Paolucci et al. proposed an ontology-based solution, in which matching of Input and Output parameters of Services are compared according to the hierarchical concept subsumption relationships defined in an ontology tree. There are four semantic similarity grades: Exact, Subsumes, Plugln, and Fail.

In [8] Li and Horrocks uses a DAML-S based ontology and a Description Logic reasoner to compare ontology based service descriptions. They extend the degrees of match of [3] by adding an intersection match. The hybrid semantic service matchmaker FCMATCH [9] performs a combined logic-based and text similarity-based matching of monolithic service and query concepts written in OWL-DL. Similarly, the work in [10] semantically matches requested and offered parameters, modeling the matchmaking problem as one of matching bipartite graphs. In [11], Peng and Shi have replaced the match grades of [3] with fine values denoted by real number, and it is used to further rank advertisements which have been matched to the same grade. Wang et al. [12] work proposes a semantic match algorithm based on improved semantic distance. Four kinds of relations are considered. Specialization, Generalization, the binary relation and the similar relation. In [13], Liu et al. achieve a fusion with five grades of matching, in precise, a collaboration of syntactic and semantic matching, as well as considering Qos and other dependency features. The OWLS-MX [14] matchmaker performs hybrid semantic matching that complements logic based reasoning with syntactic IR based similarity metrics. OWL-SLR [15] provides retrieval of Web services based not only on subsumption relationships, but also exploits the structural information of OWL ontologies. According to the work of [16], in the first phase, two web services` input/output parameters are compared semantically. In the second phase, the type of the input/output parameters of the web services is compared. In the third phase, the matching rate of the semantic web services is computed based on the results of the first and second phase. Zhang et al. [17] proposed a way to precisely compute the similarity of concepts after classifying the services into five different matchmaking levels. The weighted semantic distance and the common features of concepts are considered in similarity computation. Cai et al. [18] proposed a semantic matchmaker. But it focuses only on manufacturing domain. The similarity matching assumes either the total number of super classes subsuming the compared concepts or the total number of subclasses subsumed by the compared concepts in a shared ontological taxonomy. In addition, constraint reasoning is performed to deal with the more complex matches.

All the above mentioned works expect the user input to be given in the form of a standard service description. Less support is provided for accepting user request in natural language. So our work proposes a matchmaking algorithm which accepts user input in natural language and performs both functional matching and text-based matching so that the recall rate is improved and is more user-friendly compared to the existing approaches.

IV. PROPOSED ALGORITHM

Our discovery process aims to enable efficient search for appropriate Web services based on a user query. In our proposed discovery framework, we suppose that there is a set a Web services described in OWL-S language and is published by services providers. In order to address one of the major limitations of existing approaches, which expects the user request to be specified as an OWL-S query, our discovery framework permits user query to be expressed in natural language. Then, we perform query pre-processing using NLP techniques like stop word removal, stemming and query enhancement. The final process in the framework is semantic matchmaking. It is based on functional similarity (Input / Output concepts) and textual similarity. The architecture of our proposed work is illustrated in fig. 1.

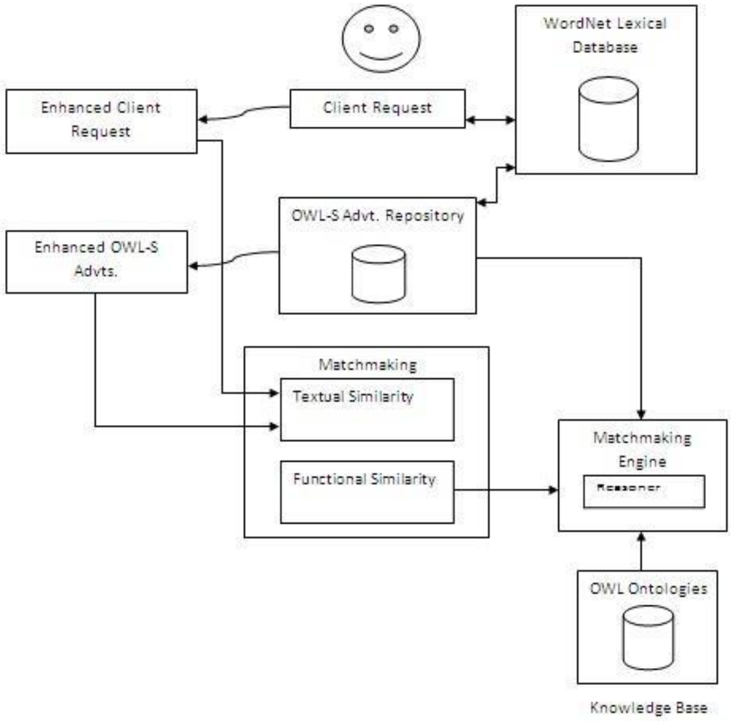

Fig. 1. Architecture of Proposed System

The major phases involved in our work are:

- User request enhancement
- Semantic matching
    - Functional similarity
    - Textual similarity

A. User request enhancement

From the service requester point of view, our system offers a simple graphical user-friendly interface to input his/her query in natural language. It also provides facilities for the user to enter the input and output parameters of the needed service. The framework enhances the user query by performing some preprocessing based on NLP techniques. The main tasks performed at this stage are stop word removal, stemming and enhancing user request by adding synsets of key terms using WordNet. Fig. 2 illustrates the various activities that are carried out during user query enhancement.

1) Service Request Pre-processing

The service request (SR) is parsed and pre-processed. Pre-processing involves the following steps:

- *Tokenization*

The user query is first partitioned into a list of words. This process includes: translation of upper case characters into lower case, punctuation removal, and white space is used as term delimiters.

- *Stop word removal*

Stop word removal aims to reduce words which act poorly as index terms. For example, those words can be "a", "the", "and" etc. An external stop word list is used to filter out those words.

- *Stemming*

Stemming is a process to replace words with their root or stem forms by removing affixes (suffixes or prefixes). Words such as: *intersected, intersecting, intersection* is stemmed as *intersect*. This process reduces not only the variety of words, but also the computational cost. The Porter Stemming Algorithm [19] is used to conduct the stemming process. It uses the WordNet lexicon dictionary.

- *Query Enhancement*

The outcome of this pre-processing is a term vector. This vector is enhanced by identifying the synsets from WordNet lexicon.

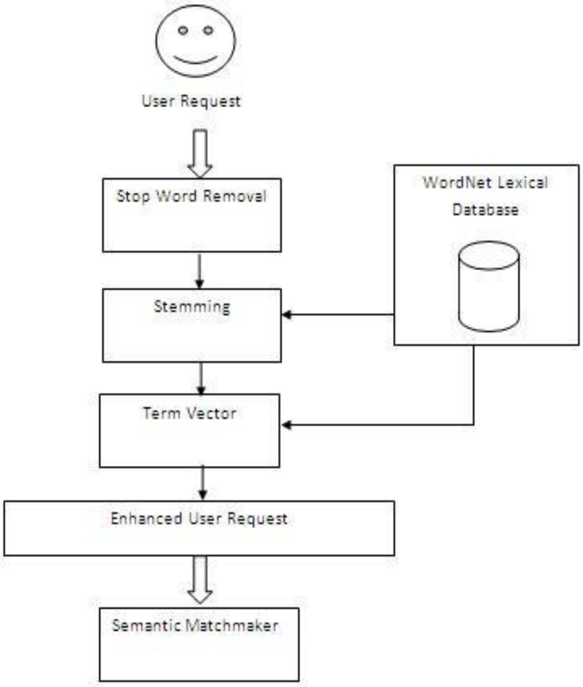

Fig. 2. Enhancement of User request

B. Semantic Matchmaking

The matchmaking process involves two activities namely: performing functional similarity [17] using the input and output parameters of the enhanced user request and the available OWL-S services and textual similarity based on enhanced user request and enhanced text description of OWL-S services [20].

1) Functional Similarity

Paolucci's algorithm is dependent on the order in which concepts are defined in the Query, which is a major drawback of this approach. Due to this drawback, the algorithm may generate false positive or false negative results. To solve this problem, another approach towards semantic matchmaking is introduced. It makes use of bipartite graph matching to produce the required match. The bipartite graph algorithm also introduces a different set of rules for match between concepts. The Plug-in and Subsume levels are interchanged in their degree of match. The assumption of Paolucci, "if an advertiser advertises a concept, it would provide all the immediate subtypes of that concept" is dropped. Hence, if the query concept is subsumed by the advertisement concept a Subsume is returned while if the query concept subsumes the advertisement concept Plug-in is returned. Plug-in is still ranked higher than a Subsume match. The algorithm for matchmaking of output parameters is given below:

```
Algorithm: PROCEDURE match (outA, outQ)
        Step 1: if outA = outQ then
                return Exact
        Step 2: else if outQ superclass of outA then
                return Plug-in
        Step 3: else if outQ subsumes outA then
                return Plug-in
        Step 4: else if outA subsumes outQ then
                return Subsumes
        Step 5: else
                return Fail
        Step 6: end if
```

Matching:-

A matching of a bipartite graph G = (V, E) is a sub graph G' = (V, E') such that no two edges e1, e2 in E' share the same vertex. Let the set of output concepts for query and advertisement be Q and A. We will construct a graph G = (Q+A, E) which has one vertex corresponding to each concept in query and advertisement. If there exists a degree of match (≠ Fail) between a concept v1 belonging to Q and a concept v2 belonging to A, then we define an edge (v1, v2) with weight as the degree of match. We need a matching in which all the output concepts of Q are matched with some concept of A. If such a matching exists, we would say that the advertisement and the query match. If there exist multiple such matching, we will choose the one which is optimal. For, this we would assign different numerical weights to edges with different degrees of match. Let us suppose, we assign minimum weight to exact, then Plug-in and then subsumes. Let max (w_i) be the maximum weighted edge in the matching. An optimal matching in this case would be a complete matching with minimum max (w_i) [10].

2) Textual similarity

Sometimes logical matching fails because of the lack of the logical relationship between a pair of concepts in a domain ontology. Also, users may not be expert enough to specify their requirements in terms of OWL-S query. In such situations, totally relying on functional (signature) similarity measures would fail to discover two similar web services. Therefore, in order to get better results, specification (i.e. textual description) similarity measures should be combined with signature similarity measures. OWL-S service profile provides the signatures using <hasInput> and <hasOutput> tags and specification description using <textDescription> tag of a web service. Usually, a textual description of a web service provides a brief functional description of what it is. The information from the <textDescription> tag is first retrieved. The following steps are carried out: converting the textual description into a vector of terms, removing stop-words, stemming, and enhancement of the term vector using WordNet. The above steps are similar to what is done for enhancing user's request. Then the similarity is calculated between the enhanced user query vector and the enhanced text description vector terms. Jaccard similarity [21] measure is used to calculate the similarity between terms in the Query and in the Advertisement. Based on this measure, the retrieved services are sorted. The text similarity is calculated as follows:

$$\text{Text\_Sim (Advt, Query)} = (\text{Advt}_T \text{ U Query}_T) / ((\text{Advt}_T + \text{Query}_T) - (\text{Advt}_T \text{ U Query}_T)) \qquad \text{eq. (1)}$$

Finally, the compound similarity of two services S_1 and S_2 is calculated by the linear combination of bipartite matching (functional similarity) and text-based similarity as follows:

$$\text{Sim } (S_1, S_2) = (\text{Functional Similarity } (S_1, S_2) + \text{Text Based Similarity } (S_1, S_2)) / 2 \qquad \text{eq. (2)}$$

V. EXPERIMENTAL RESULTS

The following algorithms were implemented in Java in order to compare their correctness and performance:

- Paolucci's algorithm
- Bipartite matching algorithm
- Proposed user-friendly bipartite with text-based matching

A. Evaluation Setup

For semantic matchmaking, it is assumed that advertisements are defined in OWL-S format. The atomic OWL-S services are selected from the OWLS-TC v2 collection [22], which is a publicly available collection of OWL-S services used to evaluate and compare different matchmaking algorithms. It comprises 1007 services which uses reference ontology with 4694 concepts from 7 different domains. The Pellet reasoner [23] is used to classify the loaded ontologies. The Jena API is used for reasoning concept relationships. The input parameter, output parameter and text-description are considered for the matchmaking process.

For example, our algorithms were tested with a user query for Hospital_Investigating_Service which has one input and one output parameter namely Hospital and Investigating respectively and its description is "Returns investigating of Hospital". Our proposed bipartite with text-based matching approach retrieves 8 more services compared to existing Bipartite matching, out of which 3 services are related to the user request.

The performance of the various matchmaking algorithms can be measured based on their recall and precision rates. Experimental results show that our proposed work has a better recall rate. The precision and recall rates can be mathematically calculated as shown below:

$$\text{Precision} = \frac{\text{Number of relevant Web services retrieved}}{\text{Total number of retrieved Web services from the dataset}} \qquad \text{eq. (3)}$$

$$\text{Recall} = \frac{\text{Number of relevant Web services retrieved}}{\text{Total number of relevant Web services in the dataset}} \qquad \text{eq. (4)}$$

Overall results listing the precision and recall rates are shown in Table I. This shows that the Bipartite matching algorithm significantly improves the retrieval of relevant services as compared to Paolucci's approach. Our proposed approach performs even better in terms of recall rate and also it is evident that it is more user-friendly.

TABLE I
PRECISION AND RECALL RATES OF THE RESULT

Method	Precision	Recall
Paolucci Matching	48.6%	82.3%
Bipartite Matching	80%	64.6%
Proposed Matching	76%	88.2%

Graphical representation of the three different matchmaking algorithms namely Paolucci's algorithm, bipartite matching and our proposed user-friendly semantic-based algorithms are shown in fig. 3.

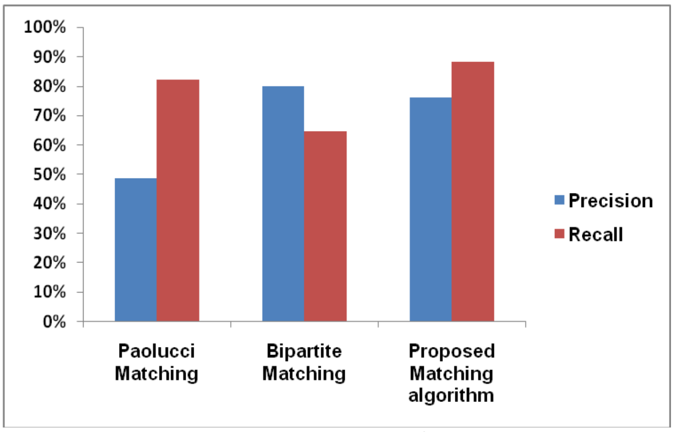

Fig. 3 Precision and Recall Rates of the three algorithms

VI. Conclusion and Future Work

The work proposed in this paper provides an approach for improved and user-friendly discovery of Web services. We give more importance on the fact that, since users may not have much knowledge about Web service related technologies and implementation details, a discovery framework that accepts a user query expressed in natural language as input is needed. Our proposed framework presents a discovery mechanism that enables Web service discovery based on keywords written in natural language. Also the experimental results show the effectiveness of our proposed approach. As a scope for future work, since at present our work is limited to atomic processes, this can be extended to deal with composite processes.

References

1. http://uddi.org/.
2. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S., “Web Services Description Language (WSDL) 1.1”, Retrieved from http://www.w3.org/TR/2001/NOTE-wsdl-20010315, 2001.
3. Paolucci, M., Kawamura, T., Payne, T.R. and Sycara, K., “Semantic matching of Web services capabilities”, Proceedings of 1st International Semantic Web Conference, Italy, pp. 333–347, 2002.
4. Antoniou, G. et al., “Web Ontology Language: OWL. Handbook on Ontologies in Information Systems”, 2003.
5. David Martin et al., “OWL-S: Semantic Markup for Web Services”, Retrieved from http://www.w3.org/Submission/OWL-S”, 2004.
6. http://wordnet.princeton.edu/.
7. “JENA: Java Framework for Building Semantic Web Applications”, Retrieved from http://jena.sourceforge.net/.
8. Lei Li and Ian Horrocks, “A software framework for matchmaking based on semantic web technology”, Proceedings of the 12th Int. Conf. on WWW, ACM New York, pp. 331-339, 2003.
9. D. Bianchini, V. D. Antonellis, M. Melchiori, D. Salvi, “Semantic-enriched service discovery”, Proceedings of IEEE ICDE 2nd International Workshop on Challenges in Web Information Retrieval and Integration (WIRI06), Atlanta, Georgia, USA, 2006.
10. Umesh Bellur, Roshan Kulkarni, “Improved Matchmaking Algorithm for Semantic Web Services Based on Bipartite Graph Matching”, IEEE International Conference on Web Services, pp. 86-93, 2007.
11. H. Peng, Z. Shi, L. Chang, and W. Niu, "Improving Grade Match to Value Match for Semantic Web Service Discovery," The IEEE International Conference on Natural Computation (ICNC 2008), IEEE Computer Society, Jinan, China, pp. 232 -236,October 2008.
12. Gongzhen Wang, Donghong Xu, Yong Qi, Di Hou, “A Semantic Match Algorithm for Web Services Based on Improved Semantic Distance”, IEEE 4th International Conference on Next Generation Web Services Practices, 2008.
13. M. Liu, Q. Gao, W. Shen, Q. Hao, and I. Van, "A Semantic-Augmented Multi-level Matching Model of Web Services", Service Oriented Computing and Applications, vol. 3, issue 3, pp. 205-215, September 2009.
14. K. Klusch M, Fries B, Sycara, “OWLS-MX: A Hybrid Semantic Web service matchmaker for OWL-S services”, Journal of Web Semantics: Science, Services and Agents on the WWW, pp. 121–133, 2009.
15. Georgios Meditskos and Nick Bassiliades, “Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S”, IEEE Transaction on Knowledge and Data Engineering, vol. 22, no.2, 2010.
16. Golsa Heidary, Kamran Zamanifar, Naser Nematbakhsh, “A Three phase Semantic Web Matchmaker”, International Journal of Smart Home Vol. 4, No.3, July, 2010.
17. Yang Zhang, Fagui Liu, Nan Zhang, “Toward Fine Grained Matchmaking of Semantic Web Services Based on Concept Similarity”, Journal of Information & Computational Science. 8: 2, pp. 377-384, 2011.

18. M. Cai, W. Y. Zhang, and K. Zhang, "ManuHub: A Semantic Web System for Ontology-Based Service Management in Distributed Manufacturing Environments", IEEE Trans. on Systems, Man & Cybernetics-Part A: Systems & Humans, Vol. 41, No. 3, May 2011.
19. http://ccl.pku.edu.cn/doubtfire/NLP/Lexical_Analysis/Word_Lemmatization/Porter/ Porter Stemming Algorithm.htm.
20. Yasser Ganjisaffar, Hassan Abolhassani, Mahmood Neshati, "A Similarity Measure for OWL-S Annotated Web Services", Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, 2006.
21. trac.research.cc.gatech.edu/ccl/export/184/SecondMindProject/SM/SM.WordNet/Paper/WordNetDotNet_Semantic_Similarity.pdf.
22. "OWL-S Service Retrieval Test Collection. Version 2.1", Retrieved from http://projects.semwebcentral.org /projects/owls-tc/.
23. Sirin, E. et al., "Pellet: An OWL DL Reasoner", Journal of Web Semantics, 2005.

BIOGRAPHY

Mrs. M.Deepa Lakshmi is a Research Scholar in Noorul Islam University, Kumaracoil. She is working as Associate Professor in the Department of Computer Science and Engineering at PSN College of Engineering and Technology, Tirunelveli District, Tamil Nadu with more than 13 years of teaching experience. Her areas of interest are web technologies and mobile computing.

Dr. Julia Punitha Malar Dhas received PhD degree in Computer Science in 2010 from Dr. M.G.R. University. She received her M.E. (CSE) degree from Madras University and B.E. degree from Bharathiyar University. She is currently a Professor and Head in the Department of CSE at Noorul Islam University, Kumaracoil, K.K.District, Tamil Nadu with more than 14 years of teaching experience. Her research interests include Wireless Networks, Mobile computing.